# Explaining Query Answers in Lightweight Ontologies: The *DL-Lite* Case

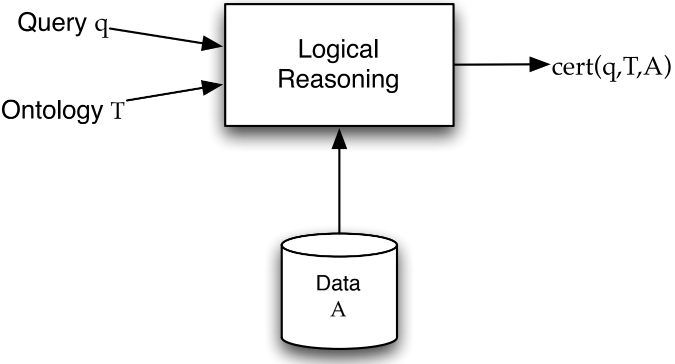
Giorgio Stefanoni
Supervisor: Prof. T. Eiter
Co-Supervisors: Dr. M. Ortiz  Dr. M. Šimkus
Scientific Advisor: D. Calvanese

Department of Computer Science
University of Oxford, UK

February 20, 2012

# Outline

# Query Answering in Description Logics



# Conjunctive Queries

- Formal counterpart of Select-Project-Join Queries in RA.

$$q(\vec{x}) \leftarrow \exists \vec{y}.\psi(\vec{x}, \vec{y})$$

- $\psi$ is a conjunction of atoms over constants and variables of the form:

$$A(t) \qquad R(t, t')$$

- A Union of CQs (UCQ) is a disjunction of CQs, corresponding to a union of SPJs.

# *DL-Lite*$_\mathcal{A}$

- Lightweight Description Logic tailored for accessing large data sources.
- Concepts and roles model set of objects and relationships among them.

$$C \rightarrow A \mid \exists R \qquad R \rightarrow P \mid P^-$$

- A *DL-Lite*$_\mathcal{A}$ ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ is composed of:

  TBox $\mathcal{T}$ Specifying constraints at the conceptual level.

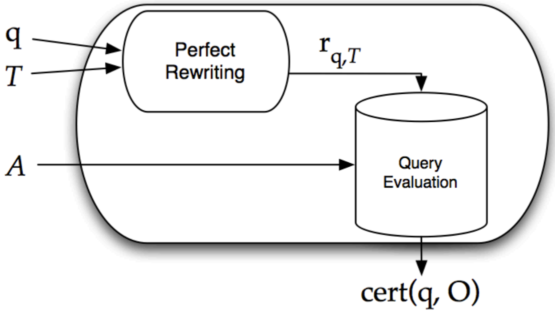
$$C \sqsubseteq D \qquad C \sqsubseteq \neg D \qquad (\text{funct } R)$$

$$R_1 \sqsubseteq R_2 \qquad R_1 \sqsubseteq \neg R_2$$

  ABox $\mathcal{A}$ Specifying the facts that hold in the domain.

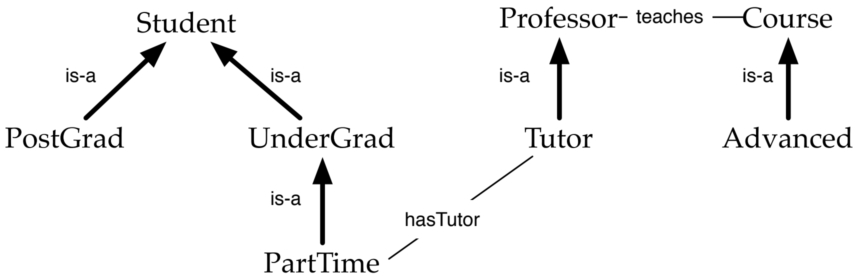
$$A(b) \qquad P(a, b)$$

# FO-Rewritability




The perfect reformulation *embeds* terminological information into $r_{q,\mathcal{T}}$.

# Mock Ontology




| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $PostGrad$ | $\sqsubseteq$ | $Student$ | $Tutor$ | $\sqsubseteq$ | $Professor$ | $Advanced$ | $\sqsubseteq$ | $Course$ |
| $UnderGrad$ | $\sqsubseteq$ | $Student$ | $\exists hasTutor$ | $\sqsubseteq$ | $PartTime$ | $\exists teaches$ | $\sqsubseteq$ | $Professor$ |
| $UnderGrad$ | $\sqsubseteq$ | $\neg Postgrad$ | $\exists hasTutor^-$ | $\sqsubseteq$ | $Tutor$ | $\exists teaches^-$ | $\sqsubseteq$ | $Course$ |
| $PartTime$ | $\sqsubseteq$ | $Student$ | | | | | | |

# Query (1)

University Database:

$$teaches(craig, SWT)$$

$$hasTutor(peter, craig)$$

Query:

$$q_1(x) \leftarrow Professor(x)$$

$$\mathsf{cert}(q_1, \mathcal{T}, \mathcal{A}) = \{craig\}$$

- In the database there is no information on Professors, how did the system retrieve the answer?

# Query (2)

University Database:

$$teaches(craig, SWT)$$

$$hasTutor(peter, craig)$$

Query:

$$q_2(x) \leftarrow teaches(x, y), Advanced(y), hasTutor(z, x)$$

$$\mathsf{cert}(q_2, \mathcal{T}, \mathcal{A}) = \emptyset$$

- Why is $craig$ not an answer?
- Is $SWT$ an $Advanced$ course?
- Does $craig$ teach a course not listed in the database?

# Outline

# Aim

Provide explanations of the following form:

| Axiom | Reason |
|---|---|
| $hasTutor(peter, craig)$ | $craig$ tutors |
| $\exists hasTutor^{-} \sqsubseteq Tutor$ | $craig$ is a Tutor |
| $Tutor \sqsubseteq Professor$ | $craig$ is a Professor |

# Aim

Provide explanations of the following form:

| Axiom | Reason |
|---|---|
| $hasTutor(peter, craig)$ | $craig$ tutors |
| $\exists hasTutor^{-} \sqsubseteq Tutor$ | $craig$ is a Tutor |
| $Tutor \sqsubseteq Professor$ | $craig$ is a Professor |

Strategy: Gather information on how TBox axioms are used to generate the perfect reformulation.

# $PerfectRef(q, \mathcal{T})$ in a (non-rigorous) Nutshell

- $\{q\} \subseteq PerfectRef(q, \mathcal{T})$.
- For each $r \in PerfectRef(q, \mathcal{T})$, we consider different cases:
  1. $r(x) \leftarrow Professor(x)$ and $Tutor \sqsubseteq Professor \in \mathcal{T}$. Then,

$$r'(x) \leftarrow Tutor(x)$$

# $PerfectRef(q, \mathcal{T})$ in a (non-rigorous) Nutshell

- $\{q\} \subseteq PerfectRef(q, \mathcal{T})$.
- For each $r \in PerfectRef(q, \mathcal{T})$, we consider different cases:
  1. $r(x) \leftarrow Professor(x)$ and $Tutor \sqsubseteq Professor \in \mathcal{T}$. Then,
  $$r'(x) \leftarrow Tutor(x)$$
  2. $r(x) \leftarrow hasTutor(x, y)$ and $PartTime \sqsubseteq \exists hasTutor$. Then,
  $$r'(x) \leftarrow PartTime(x)$$

# $PerfectRef(q, \mathcal{T})$ in a (non-rigorous) Nutshell

- $\{q\} \subseteq PerfectRef(q, \mathcal{T})$.
- For each $r \in PerfectRef(q, \mathcal{T})$, we consider different cases:
  1. $r(x) \leftarrow Professor(x)$ and $Tutor \sqsubseteq Professor \in \mathcal{T}$. Then,
  $$r'(x) \leftarrow Tutor(x)$$
  2. $r(x) \leftarrow hasTutor(x, y)$ and $PartTime \sqsubseteq \exists hasTutor$. Then,
  $$r'(x) \leftarrow PartTime(x)$$
  3. $r(x) \leftarrow Professor(x)$ and $\exists teaches \sqsubseteq Professor$. Then,
  $$r'(x) \leftarrow teaches(x, \_)$$

# Computing Positive Explanations

- Maintain a graph $G$ of rewritings.
  - $(r, r') \in G$ means that $r'$ has been generated from $r$.
  - Label $(r, r')$ with the axiom justifying the rewriting.
- Let $\pi$ be a match for $r \in \mathit{PerfectRef}(q_1, \mathcal{T})$ in $\mathcal{A}$ witnessing $craig$.
- **IDEA:** Traverse backwards the trace of rewritings from $r$ until $q_1$ is reached. Suitably extend $\pi$ to be a match for intervening queries.

# Example

$$q_1(x) \leftarrow Professor(x)$$

$teaches(craig, SWT)$ Database $hasTutor(peter, craig)$

# Example

$$q_1(x) \leftarrow Professor(x)$$

$$r_1(x) \leftarrow Tutor(x)$$

$teaches(craig, SWT)$ Database $hasTutor(peter, craig)$

# Example

$$q_1(x) \leftarrow Professor(x)$$

$$r_1(x) \leftarrow Tutor(x)$$

$$r_2(x) \leftarrow hasTutor(y, x)$$

$teaches(craig, SWT)$ Database $hasTutor(peter, craig)$

$\pi$ matches $x$ on $craig$ and $y$ on $peter$.

# Example

$$q_1(x) \leftarrow Professor(x)$$

$$r_1(x) \leftarrow Tutor(x)$$

$$r_2(x) \leftarrow hasTutor(peter, craig)$$

$teaches(craig, SWT)$ Database $hasTutor(peter, craig)$

$\pi$ matches $x$ on $craig$ and $y$ on $peter$.

# Example

$$q_1(x) \leftarrow Professor(x)$$

$$r_1(x) \leftarrow Tutor(craig)$$

$$r_2(x) \leftarrow hasTutor(peter, craig)$$

$teaches(craig, SWT)$ Database $hasTutor(peter, craig)$

$\pi$ matches $x$ on $craig$ and $y$ on $peter$.

# Algorithmic Solution

- Modify *PerfectRef* to maintain rewriting graph.
- At explanation time, use Dijkstra algorithm to find shortest path between generating rewriting and user query.
- Extend match on generating rewriting for intervening queries.
- Return shortest path and extended match.

# Complexity

- Dijkstra runs in $O(|V|^2)$.
- In our case, the number of vertexes is the number of conjunctive queries in $PerfectRef(q, \mathcal{T})$.
- Worst-case: a CQ $q$ admits exponentially many rewritings w.r.t. *DL-Lite*$_{\mathcal{A}}$ TBox $\mathcal{T}$.
- Our explanation algorithm runs in exponential time w.r.t. the query.
- Data-complexity is still low.

# Outline

# Query (2)

University Database:

$$teaches(craig, SWT)$$

$$hasTutor(peter, craig)$$

Query:

$$q_2(x) \leftarrow teaches(x, y), Advanced(y), hasTutor(z, x)$$

$$\mathsf{cert}(q_2, \mathcal{T}, \mathcal{A}) = \emptyset$$

# Method

- **Abductive Reasoning**: solutions are assertions to be added to the ontology leading the given tuple to be returned by the system.
- Solutions should be non-redundant: study minimality conditions!

# Abductive Reasoning

- A form of non-sequitor argument, in which

$$\Gamma \not\models B$$

but $B$ is assumed to follow from the premises.

- Solutions are set of formulae $\mathcal{E}$ such that

$$\Gamma \cup \mathcal{E} \models B$$

- Natural conditions over solutions:

  Consistency $\Gamma \cup \mathcal{E} \not\models \bot$

  Minimality $\mathcal{E}$ is minimal wrt. some criterion.

# Reasoning over Abduction Problems

1. Does there exist a (minimal) solution? (EXIST)
2. Does a formula $\alpha$ occur in all (minimal) solutions? (NEC)
3. Does a formula $\alpha$ occur in some (minimal) solution? (REL)
4. Is a set $\mathcal{E}$ of formulae a (minimal) solution? (REC)

# Query Abduction Problem

- We call $\mathcal{P} = \langle \mathcal{T}, \mathcal{A}, Q(\vec{x}), \vec{a} \rangle$ a QAP, where
  1. $\langle \mathcal{T}, \mathcal{A} \rangle$ is a *DL-Lite*$_\mathcal{A}$ ontology.
  2. $Q(\vec{x})$ is a Union of CQs.
  3. $\vec{a}$ is a tuple of constants of matching arity.
- A solution to $\mathcal{P}$ is an ABox $\mathcal{E}$ such that:
  - $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{E} \rangle$ is consistent.
  - $\vec{a} \in \text{cert}(q, \mathcal{T}, \mathcal{A} \cup \mathcal{E})$.
- We denote with $\text{expl}(\mathcal{P})$ the set of all solutions to $\mathcal{P}$.

# Properties of QAPs

$$\mathcal{P} = \langle \mathcal{T}, \mathcal{A}, Q(\vec{x}), \vec{a} \rangle$$

- If $\vec{a} \notin \text{cert}(q, \mathcal{T}, \mathcal{A})$, we call $\vec{a}$ a negative answer to $Q$ over the ontology.
- Negative answers exist only if the ontology is consistent.
- If the ontogy is inconsistent, the the QAP does have solutions.
- A solution $\mathcal{E}$ to QAP $\mathcal{P}$ can introduce constants not occurring in the ABox $\mathcal{A}$.

# Reasoning & Preference Orders

- We consider the four reasoning tasks over abductive problems under 3 different preference orders:
  - no minimality condition,
  - subset-minimality order denoted by $\subseteq$, and,
  - minimum explanation size order denoted by $\leq$.

# Query (2)

University Database:

$teaches(craig, SWT)$

$hasTutor(peter, craig)$

Query:

$$q_2(x) \leftarrow teaches(x, y), Advanced(y), hasTutor(z, x)$$

| | ABox additions: |
|---|---|
| $\leq$ | $Advanced(SWT)$ |
| $\subseteq$ | $teaches(craig, new : ALG), Advanced(new : ALG)$ |
| none | $teaches(craig, new : TOC), hasTutor(new : Ben, craig), Advanced(new : TOC)$ |

# Outline of Complexity Results

| | $\preceq$-EXIST | $\preceq$-NEC | $\preceq$-REL | $\preceq$-REC |
|---|---|---|---|---|
| none | PTime | PTime | PTime | NP |
| $\leq$ | PTime | $P_{\parallel}^{NP}$ | $P_{\parallel}^{NP}$ | DP |
| $\subseteq$ | PTime | PTime | $\Sigma_2^P$ | DP |

# Outline of Complexity Results

| | $\preceq$-EXIST | $\preceq$-NEC | $\preceq$-REL | $\preceq$-REC |
|---|---|---|---|---|
| none | PTime | PTime | PTime | NP |
| $\leq$ | PTime | $P_{\parallel}^{NP}$ | $P_{\parallel}^{NP}$ | DP |
| $\subseteq$ | PTime | PTime | $\Sigma_2^P$ | DP |

# Canonical Explanations

- If QAP $\mathcal{P} = \langle \mathcal{T}, \mathcal{A}, Q, \vec{a} \rangle$ has a solution, then there is a small solution.
- Finding a solution amounts to satisfy one of the CQs in $Q$.
- Satisfying a CQ does not require more than the number of terms contained in the query itself.
- Hence, one can find a solution by instantiating terms occurring in the query using a small number of new constants.

# Complexity of $\subseteq$-EXIST

- A minimal solution to a QAP $\mathcal{P}$ exists iff $\mathcal{P}$ has a (general) solution.

**Theorem**

*For DL-Lite$_{\mathcal{A}}$,* EXIST *is in* PTime-*complete.*

**Upper bound intuition.**

- Consider QAPs over CQs, general result for UCQs follows.
- Treat the body of the query as an ABox $\mathcal{E}$ and set $\mathcal{O} = \mathcal{O} \cup \mathcal{E}$.
- Replace each variable $x$ in $\mathcal{E}$ with a variable representative $a_x$.
- Use disjointness in $\mathcal{O}$ to enforce distinctness among constants. Thus, only variable representatives can be identified.
- Check satisfiability of the resulting ontology $\mathcal{O}$ without the UNA.

□

# Complexity of ⊆-NEC

- An assertion is ⊆-necessary iff it is necessary.

**Theorem**

*For DL-Lite$_{\mathcal{A}}$, NEC is PTime-complete.*

**Upper bound intuition.**

- We want to decide whether $A(a)$ is *necessary* for $\mathcal{P} = \langle \mathcal{O}, q, \vec{a} \rangle$.
- Check whether $A(a)$ is a consequence of $\mathcal{O}$. In case return no.
- Create $\mathcal{P}' = \langle \mathcal{O}', q, \vec{a} \rangle$ by extending $\mathcal{O}$ as follows:

$$\mathcal{T} \cup \bar{A} \sqsubseteq \neg A \qquad \mathcal{A} \cup \{\bar{A}(a)\}$$

- Check that $\mathcal{P}'$ does not admit solutions. If this is the case return yes.

□

# Complexity of $\subseteq$-REL

## Theorem

*For DL-Lite$_{\mathcal{A}}$, $\subseteq$-REL is $\Sigma_2^{\mathsf{P}}$-complete.*

## Upper bound intuition.

- We want to decide whether $A(a)$ is $\subseteq$-relevant for $\mathcal{P} = \langle \mathcal{T}, \mathcal{A}, q, \vec{a} \rangle$.
- Guess a derivation of one rewriting $r$ in $PerfetctRef(q, \mathcal{T})$.
- Guess a subset $E$ of the atoms of $r$
- Guess an instantiation $\mathcal{E}$ of the atoms in $E$.
- Check that $\mathcal{E}$ is an explanation for $\mathcal{P}$. (NP)
- Check that $\mathcal{E}$ is minimal (coNP)

□

# Complexity of ⊆-REC

**Theorem**

*For DL-Lite$_{\mathcal{A}}$, ⊆-REC is DP-complete.*

**Upper bound intuition.**

- By definition of DP.
- A language $L$ is in DP if there are two languages $L_1$ and $L_2$, resp. in NP and coNP such that:

$$L = L_1 \cap L_2$$

- Thus

$$L_1 = \{\langle \mathcal{P}, \mathcal{E} \rangle \mid \mathcal{E} \in \mathsf{expl}(\mathcal{P})\}$$
$$L_2 = \{\langle \mathcal{P}, \mathcal{E} \rangle \mid \neg\exists \mathcal{E}' \in \mathsf{expl}(\mathcal{P}) \text{ such that } \mathcal{E}' \subset \mathcal{E}\}$$
$$\subseteq\text{-REC} = L_1 \cap L_2$$

# Outline

# Conclusions

- Provide an algorithmic solution to the problem of explaining positive answers.
- Contribute with a new formalization to the problem of explaining negative answers over ontologies as an abductive task.
- For $\mathit{DL\text{-}Lite}_{\mathcal{A}}$, we study the complexity of reasoning over QAPs under minimality conditions.

# Publications

- *The Complexity of Conjunctive Query Abduction in DL-Lite*. Diego Calvanese, Magdalena Ortiz, Mantas Simkus, and Giorgio Stefanoni Proc. of the 24th Int. Workshop on Description Logics (DL 2011). Volume 745 of CEUR Electronic Workshop Proceedings, http://ceur-ws.org/. 2011.
- *The Complexity of Explaining Negative Query Answers in DL-Lite*. Diego Calvanese, Magdalena Ortiz, Mantas Simkus, and Giorgio Stefanoni. Accepted to KR2012.